

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4405884号
(P4405884)

(45) 発行日 平成22年1月27日(2010.1.27)

(24) 登録日 平成21年11月13日(2009.11.13)

(51) Int.Cl.

B 41 J 5/30 (2006.01)
G 06 F 3/12 (2006.01)

F 1

B 41 J 5/30
G 06 F 3/12Z
B

請求項の数 6 (全 16 頁)

(21) 出願番号 特願2004-275029 (P2004-275029)
 (22) 出願日 平成16年9月22日 (2004.9.22)
 (65) 公開番号 特開2006-88433 (P2006-88433A)
 (43) 公開日 平成18年4月6日 (2006.4.6)
 審査請求日 平成18年6月7日 (2006.6.7)

(73) 特許権者 000001007
 キヤノン株式会社
 東京都大田区下丸子3丁目30番2号
 (74) 代理人 100076428
 弁理士 大塚 康徳
 (74) 代理人 100112508
 弁理士 高柳 司郎
 (74) 代理人 100115071
 弁理士 大塚 康弘
 (74) 代理人 100116894
 弁理士 木村 秀二
 (72) 発明者 伊藤 順康
 東京都大田区下丸子3丁目30番2号 キ
 ャノン株式会社内

最終頁に続く

(54) 【発明の名称】描画処理回路、及び画像出力制御装置

(57) 【特許請求の範囲】

【請求項 1】

ページ記述言語で記述されたデータを受信して、該受信したデータにより記述された描画オブジェクトが有する異なる属性をそれぞれ表現する複数の属性データを含む描画オブジェクトデータを生成する描画オブジェクト生成手段と、

前記描画オブジェクト生成手段が生成した描画オブジェクトデータをビットマップデータに展開するための複数の展開手段であって、前記属性データの種類に応じた複数種類の展開処理を実行するために動的に再構成可能なプロセッサを有する複数の展開手段と、

前記展開処理を実行する前に、前記複数の展開手段の各々に割り当てられた属性データに基づいて前記割り当てられた属性データを前記複数の展開手段の各々が展開処理するのに要する処理時間を見積り、前記複数の展開手段のアイドル状態を少なくするように前記プロセッサが実行する展開処理の種類を切り替えるためのスケジューリングデータを生成するスケジューリングデータ生成手段と、

該スケジューリングデータに基づいて、前記プロセッサが実行する展開処理の種類を切り替えるよう前記プロセッサを再構成する再構成手段とを有し、

特定の描画オブジェクトデータに含まれる複数の属性データを処理する際に、前記複数の展開手段に含まれる特定の展開手段に割り当てられた属性データの展開処理が終了し、かつ前記複数の展開手段に含まれる他の展開手段に割り当てられた属性データの展開処理が終了していない場合に、

前記スケジューリングデータ生成手段は、前記他の展開手段に割り当てられた属性データ

10

20

タの展開処理を前記特定の展開手段に実行させるべく、前記スケジューリングデータを修正し、

前記再構成手段は、前記修正されたスケジューリングデータに基づいて、前記特定の展開手段が有するプロセッサが実行する展開処理の種類を前記他の展開手段が有するプロセッサが実行する展開処理の種類に切り替えるよう前記特定の展開手段が有するプロセッサを再構成することを特徴とする描画処理回路。

【請求項 2】

前記スケジューリングデータに基づいて、複数の描画オブジェクトに含まれる同一種類の複数の属性データをビットマップデータへ展開する展開処理を実行するのに要する処理時間を前記属性データの種類毎に判定する処理時間判定手段を有し、

10

前記スケジューリングデータ生成手段は、前記処理時間判定手段が判定した前記属性データの種類毎の処理時間に基づいて、前記複数の展開手段が前記複数の属性データの前記展開処理を終了する時間が各属性データで略同じとなるように、前記スケジューリングデータを修正することを特徴とする請求項 1 記載の描画処理回路。

【請求項 3】

前記描画オブジェクトに含まれる複数の属性データは、前記描画オブジェクトの描画領域を示すデータと、前記描画オブジェクトの輪郭を示すデータと、前記描画オブジェクトの色を示すデータと、前記描画オブジェクトの模様を示すデータとを含むことを特徴とする請求項 2 記載の描画処理回路。

【請求項 4】

前記複数の展開手段の各々は、ページ単位の属性データを所定単位で複数に分割した分割データ単位で前記展開処理を実行することを特徴とする請求項 1 記載の描画処理回路。

20

【請求項 5】

ページ記述言語で記述されたデータを受信してビットマップデータに展開する描画処理回路を有し、該描画処理回路で展開されたビットマップデータを出力手段に出力する画像出力制御装置であって、

前記描画処理回路が、

ページ記述言語で記述されたデータを受信して、該受信したデータにより記述された描画オブジェクトが有する異なる属性をそれぞれ表現する複数の属性データを含む描画オブジェクトデータを生成する描画オブジェクト生成手段と、

30

前記生成手段が生成した描画オブジェクトデータをビットマップデータに展開するための複数の展開手段であって、前記属性データの種類に応じた複数種類の展開処理を実行するために動的に再構成可能なプロセッサを有する複数の展開手段と、

前記展開処理を実行する前に、前記複数の展開手段の各々に割り当てられた属性データに基づいて前記割り当てられた属性データを前記複数の展開手段の各々が展開処理するのに要する処理時間を見積り、前記複数の展開手段のアイドル状態を少なくするように前記プロセッサが実行する展開処理の種類を切り替えるためのスケジューリングデータを生成するスケジューリングデータ生成手段と、

該スケジューリングデータに基づいて、前記プロセッサが実行する展開処理の種類を切り替えるよう前記プロセッサを再構成する再構成手段とを有し、

40

特定の描画オブジェクトデータに含まれる複数の属性データを処理する際に、前記複数の展開手段に含まれる特定の展開手段に割り当てられた属性データの展開処理が終了し、かつ前記複数の展開手段に含まれる他の展開手段に割り当てられた属性データの展開処理が終了していない場合に、

前記スケジューリングデータ生成手段は、前記特定の展開手段とは異なる他の展開手段に割り当てられた属性データの展開処理を前記特定の展開手段に実行させるべく、前記スケジューリングデータを修正し、

前記再構成手段は、前記修正されたスケジューリングデータに基づいて、前記特定の展開手段が有するプロセッサが実行する展開処理の種類を前記他の展開手段が有するプロセッサが実行する展開処理の種類に切り替えるよう前記特定の展開手段が有するプロセッサ

50

を再構成することを特徴とする画像出力制御装置。

【請求項 6】

前記描画処理回路は、前記スケジューリングデータに基づいて、複数の描画オブジェクトに含まれる同一種類の複数の属性データをビットマップデータへ展開する展開処理を実行するのに要する処理時間を前記属性データの種類毎に判定する処理時間判定手段を有し
、

前記スケジューリングデータ生成手段は、前記処理時間判定手段が判定した前記属性データの種類毎の処理時間に基づいて、前記複数の展開手段が前記複数の属性データの前記展開処理を終了する時間が各属性データで略同じとなるように、前記スケジューリングデータを修正することを特徴とする請求項 5 記載の画像出力制御装置。

10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は描画処理回路、及び画像出力制御装置、例えば、ホストコンピュータから印刷ジョブとしてPDLなどのObject記述言語で記述されたデータを受信し、1ページ分の前記データを複数のBandに分割してBand単位ごとの中間データに変換し、前記中間データに含まれている描画Objectの各属性に応じてbitmapへ展開する描画処理回路、及び画像出力制御装置に関するものである。

【背景技術】

【0002】

20

以下、本明細書では、画像出力制御装置として画像を印刷する印刷手段への画像出力を制御する印刷制御装置に基づいて説明する。しかし、本発明は表示手段への画像出力の制御においても適用でき、それらも本発明に含まれる。

【0003】

本発明に関連する描画処理技術としては、例えば特許文献1のような技術が一般的であった。特許文献1では、印刷データから生成される中間データと、印刷出力を行う出力手段の属性（プリンタエンジンがカラーか白黒かなど）とに基づいて、再構成制御手段によって、中間データを出力手段の仕様に合わせた描画データ(bitmap)に展開するよう展開処理手段の再構成を行っている。

【0004】

30

又、本出願人が先に出願した特許文献2には、中間データに含まれている描画Objectの各属性に応じてbitmapへ展開する場合に、各属性のデータをDMAで短時間に読み出すのが好適なDL(Display List)構造やその手順が示されている。

【特許文献1】特開平11-105371号公報

【特許文献2】特開2003-241917号公報

【発明の開示】

【発明が解決しようとする課題】

【0005】

しかしながら、特許文献1における再構成は、複数のハードウェアリソースを設けてこの固定したリソースを所望の展開処理となるように、選択あるいは接続を変えるものであり、少なくとも1つのObjectを描画する間の展開処理手段の構成は固定されたものである。同様に、特許文献2においても、DMAで読み出した各属性のデータはそれぞれ専用の固定データ生成部で展開される構成である。従って、各Objectの描画において各属性の展開処理の時間が変化しても、極端な場合は展開処理の無い属性がある場合でも、常に最も時間がかかる属性の展開処理に処理時間が拘束されてしまう。また、展開処理の無い属性のハードウェアリソースはその間、アイドル状態となるので、ハードウェアリソースの無駄が増えることになる。

40

【0006】

本発明は、上記従来技術の問題点に鑑み、Objectベースの出力データの属性に応じてハードウェアリソースを再構成できるよう構成し、その再構成のスケジューリング、更に動

50

的スケジューリングを行なうことによって、ハードウェアリソースの無駄を減らし、描画処理を高速に行なう描画処理回路、及び印刷制御装置を提供する。

【課題を解決するための手段】

【0007】

上記課題を解決するために、本発明の描画処理回路は、ページ記述言語で記述されたデータを受信して、該受信したデータにより記述された描画オブジェクトが有する異なる属性をそれぞれ表現する複数の属性データを含む描画オブジェクトデータを生成する描画オブジェクト生成手段と、前記描画オブジェクト生成手段が生成した描画オブジェクトデータをビットマップデータに展開するための複数の展開手段であって、前記属性データの種類に応じた複数種類の展開処理を実行するために動的に再構成可能なプロセッサを有する複数の展開手段と、前記展開処理を実行する前に、前記複数の展開手段の各々に割り当てられた属性データに基づいて前記割り当てられた属性データを前記複数の展開手段の各々が展開処理するのに要する処理時間を見積り、前記複数の展開手段のアイドル状態を少なくするように前記プロセッサが実行する展開処理の種類を切り替えるためのスケジューリングデータを生成するスケジューリングデータ生成手段と、該スケジューリングデータに基づいて、前記プロセッサが実行する展開処理の種類を切り替えるよう前記プロセッサを再構成する再構成手段とを有し、特定の描画オブジェクトデータに含まれる複数の属性データを処理する際に、前記複数の展開手段に含まれる特定の展開手段に割り当てられた属性データの展開処理が終了し、かつ前記複数の展開手段に含まれる他の展開手段に割り当てられた属性データの展開処理が終了していない場合に、前記スケジューリングデータ生成手段は、前記他の展開手段に割り当てられた属性データの展開処理を前記特定の展開手段に実行させるべく、前記スケジューリングデータを修正し、前記再構成手段は、前記修正されたスケジューリングデータに基づいて、前記特定の展開手段が有するプロセッサが実行する展開処理の種類を前記他の展開手段が有するプロセッサが実行する展開処理の種類に切り替えるよう前記特定の展開手段が有するプロセッサを再構成することを特徴とする。10

【0008】

前記スケジューリングデータに基づいて、複数の描画オブジェクトに含まれる同一種類の複数の属性データをビットマップデータへ展開する展開処理を実行するのに要する処理時間を前記属性データの種類毎に判定する処理時間判定手段を有し、前記スケジューリングデータ生成手段は、前記処理時間判定手段が判定した前記属性データの種類毎の処理時間に基づいて、前記複数の展開手段が前記複数の属性データの前記展開処理を終了する時間が各属性データで略同じとなるように、前記スケジューリングデータを修正する。また、前記描画オブジェクトに含まれる複数の属性データは、前記描画オブジェクトの描画領域を示すデータと、前記描画オブジェクトの輪郭を示すデータと、前記描画オブジェクトの色を示すデータと、前記描画オブジェクトの模様を示すデータとを含む。また、前記複数の展開手段の各々は、ページ単位の属性データを所定単位で複数に分割した分割データ単位で前記展開処理を実行する。30

【0009】

又、本発明の画像出力制御装置は、ページ記述言語で記述されたデータを受信してビットマップデータに展開する描画処理回路を有し、該描画処理回路で展開されたビットマップデータを出力手段に出力する画像出力制御装置であって、前記描画処理回路が、ページ記述言語で記述されたデータを受信して、該受信したデータにより記述された描画オブジェクトが有する異なる属性をそれぞれ表現する複数の属性データを含む描画オブジェクトデータを生成する描画オブジェクト生成手段と、前記生成手段が生成した描画オブジェクトデータをビットマップデータに展開するための複数の展開手段であって、前記属性データの種類に応じた複数種類の展開処理を実行するために動的に再構成可能なプロセッサを有する複数の展開手段と、前記展開処理を実行する前に、前記複数の展開手段の各々に割り当てられた属性データに基づいて前記割り当てられた属性データを前記複数の展開手段の各々が展開処理するのに要する処理時間を見積り、前記複数の展開手段のアイドル状態40

50

を少なくするように前記プロセッサが実行する展開処理の種類を切り替えるためのスケジューリングデータを生成するスケジューリングデータ生成手段と、該スケジューリングデータに基づいて、前記プロセッサが実行する展開処理の種類を切り替えるよう前記プロセッサを再構成する再構成手段とを有し、特定の描画オブジェクトデータに含まれる複数の属性データを処理する際に、前記複数の展開手段に含まれる特定の展開手段に割り当てられた属性データの展開処理が終了し、かつ前記複数の展開手段に含まれる他の展開手段に割り当てられた属性データの展開処理が終了していない場合に、前記スケジューリングデータ生成手段は、前記特定の展開手段とは異なる他の展開手段に割り当てられた属性データの展開処理を前記特定の展開手段に実行させるべく、前記スケジューリングデータを修正し、前記再構成手段は、前記修正されたスケジューリングデータに基づいて、前記特定の展開手段が有するプロセッサが実行する展開処理の種類に切り替えるよう前記特定の展開手段が有するプロセッサを再構成することを特徴とする。

【0010】

前記描画処理回路は、前記スケジューリングデータに基づいて、複数の描画オブジェクトに含まれる同一種類の複数の属性データをビットマップデータへ展開する展開処理を実行するのに要する処理時間を前記属性データの種類毎に判定する処理時間判定手段を有し、前記スケジューリングデータ生成手段は、前記処理時間判定手段が判定した前記属性データの種類毎の処理時間に基づいて、前記複数の展開手段が前記複数の属性データの前記展開処理を終了する時間が各属性データで略同じとなるように、前記スケジューリングデータを修正する。

【発明の効果】

【0014】

本発明によれば、特定の描画オブジェクトデータに含まれる複数の属性データの展開処理の処理状況に応じた適切なタイミングで複数の展開手段に含まれる特定の展開手段が有するプロセッサが実行する展開処理の種類を切り替えて、特定の描画オブジェクトデータの展開処理に要する時間を短くすることができる。

【発明を実施するための最良の形態】

【0015】

以下に、本発明の最良の実施形態として、Objectベースの印刷データの属性に応じてハードウェアリソースを再構成できるよう構成し、その再構成のスケジューリング、更に動的スケジューリングを行なうことの可能な描画処理回路とそれを有する画像出力制御装置を詳細に説明する。尚、前述の如く、本実施形態では、画像出力制御装置として画像を印刷する印刷手段への画像出力を制御する印刷制御装置に基づいて説明するが、本発明は表示手段への画像出力の制御においても適用でき、それらも本発明に含まれる。

【0016】

<本実施形態の画像出力制御装置の構成例>

図1は、本実施形態の画像出力制御装置を含む印刷装置の構成例を示す図である。例えば印刷装置の例としてはレーザビームプリンタの場合に対応する。

【0017】

図1において、101は、図示していないホストコンピュータより転送された印刷コマンド・データを受信するHost_I/F_Logicである。このHost_I/F_Logic 101は、DRAM_cont.102aを介して、受信された印刷コマンド・データをDRAM102へ格納する。尚、DMA_cont.102bは、DRAM102に格納されたDLをRasterize_mod.107に転送する場合などに使用される。

【0018】

106はCPUであり、後述するROM105に格納されたプログラムにしたがって印刷装置本体の全制御を行う演算・制御部である。CPU106は、各構成要素間の処理手順や同期制御、あるいは共通バスの制御(DMA_cont.と協働)を行なう。特に、CPU106はRasterize_mod.やShipping_mod.やVideo_I/F_LogicのY M C K プレーンに対する同期制御を行なう。尚、CPU106がRasterize_mod.107やData_compress109や拡大/縮小_mod.110やData_uncompress111な

10

20

30

40

50

どの処理を行なっても良いが、CPU106に負荷が集中して処理の高速化を妨げるので、Rasterize_mod.107の各planeそれぞれに、あるいはData_compress109や拡大／縮小_mod.110やData_uncompress111に共通、又はそれぞれに独立したCPUを有するのが望ましい。特に、本例では、Rasterize_mod.107の各planeそれが独立したCPUを有して、他のplaneとは独立に展開処理を行なう（尚、同じObjectのplane間の同期はCPU106が制御する）。

【0019】

102はDRAMで、ホストコンピュータからのコマンドおよびデータ等の受信データを一時的に蓄えるpageメモリである。あるいは前述のROM105内に記述されたプログラムに従って生成されるDLを格納するMainメモリとして使用される。また、Font cache領域として使用される。さらに前記の用途の他にCPU106が各プログラムを実行する際のワークエリアとしても使用される。105は、CPU106の制御プログラム、あるいは後述する図3及び図6に示すフローチャートによる一連の制御を実現するためのプログラムデータ、Fontデータなどを格納しているROMである。尚、図3及び図6に示すフローチャートはRasterize_mod.107の各planeの制御を示すものである。この制御はROM105から読み出しながら動作するのではなく、Rasterize_mod.107の各plane（以下の図2参照）が有するRAM（Config._control202の制御やConfig._data_memory210と共有でもよい）にロードして実行されるのが望ましい。また、DMA_cont.105aは、ROM105に格納されたFontデータの読み出しや、Rasterize_mod.107への制御プログラムのロードなどで使用される。

【0020】

103はHard_Disk、103aはHard_Disk_I/F_Logic、104はNV(non-volatile)RAM、104aはNVRAM_I/Fである。Hard Disk103及びNVRAM104は不揮発性の書換可能なメモリであり、大容量のデータや変更可能なプログラムなどが格納される。

【0021】

107は描画処理を実行する4つのplaneに分かれたRasterize_mod.である。Rasterize_mod.107は、DRAM102から読み込んだDLをROM105に格納されたプログラム（あるいはRasterize_mod.107にロードされたプログラム）に従ってbitmapデータへ展開し、Band Memory108へ格納する。109は前記のRasterize_mod.107が展開したデータを圧縮するData_compress_moduleである。110は縮小印字や文字の拡大処理を行うための拡大／縮小moduleである。111はBand Mem.108に格納された圧縮データを伸張するためのData_uncompress_moduleである。

【0022】

また、112は、カラーの4-Drumエンジンの紙搬送に対応して画像データを前記Band Mem.108から読み出して、video_I/F_logic114へ出力するための同期回路（Shipping_mod.）である。113はプリンタエンジンを制御するcommand/statusの制御信号である。114,115はプリンタエンジンとそのI/F_logicである。116,117はプリンタの表示パネルとそのI/F_logicである。

【0023】

<本実施形態の描画処理例>

次に、本実施形態の描画処理における、Object描画処理での属性の種類と、DL(Display List)の構成例、出力画像との関係について説明する。但し、本発明は描画処理におけるハードウェアリソースの再構成、その再構成のスケジューリング、更に動的スケジューリングに特徴があり、Object上が処理での属性の種類や、DLの構成例等に限定されるものではない。

【0024】

（本実施形態における属性例）

本実施形態の描画回路においては、PDLを中間コード(DL)へ変換し、この中間コードが有する以下の4つの属性データを元にして描画イメージを生成するものとする。

【0025】

第1の属性は、描画オブジェクトの1つ1つについて印刷イメージ上で占有する領域データを示す属性データである。第2の属性は、多角形、円などの図形や文字の輪郭に代表

10

20

30

40

50

されるような ForegroundとBackgroundとの境界を表す属性データである。第3の属性は、描画オブジェクトを塗りつぶす模様に代表されるようなForegroundの模様（あるいはBackgroundの色or模様）を示す属性データである。第4の属性は、その塗りつぶす色を示すForegroundの色（あるいはBackgroundの色or模様）を示す属性データである。

【0026】

ここで、Foregroundの色とは描画対象の主体の色を意味しており、文字などを描画する場合に文字自身に付けられる色を指している。Backgroundとは描画プレーン1つで2値データを扱う場合には描画を行うプレーンのハッティングパターンとなり、多値データを扱う場合にはフォアグラウンドの色を意味している。

【0027】

前記4種類の属性に基づくデータは、描画展開後に印刷する背景のイメージデータの上にそれぞれを重ねあわせる処理を行った後に、Band_Memory108へ格納される。図5は、あるオブジェクトについての各属性と描画処理の一例の概略を示したものである。

【0028】

（本実施形態におけるDLの構成例）

図9に、座標情報と描画objectの属性とを示した中間データ(display list)のコード例と構成例を示す。

【0029】

各リスト中のaddr._指定は、データの管理上、pointer形式の間接指定になっている。例えばリスト中の"obj._addr._pointer#" (#: 1,2,3,...)は各描画objectのHeaderを格納しているMemory_addr.を示すaddr._pointerである。ObjectのHeaderは、描画objectの種類（文字、図形、イメージ、背景データ、色データなど）を示すFunction_codeと幅・高さのパラメータ、及び、文字の圧縮code本体を格納しているMem.の先頭Addressを示すaddr._pointerから構成されている。

【0030】

図9においては、Object_No.1とObject_No.2とがリストされている。

【0031】

第1行から第7行まではObject_No.1の共通の属性である、領域及び形状（外形、輪郭）に係る情報である。第8行から第10行はObject_No.1のシアン（C）の属性、色（Color_Gradation）及び模様（Fill_Pattern）に係る情報である。第11行から第13行はObject_No.1のイエロー（Y）の属性、色及び模様に係る情報である。Object_No.1にはマゼンタ（M）とブラック（K）の情報は無い。

【0032】

第14行から第20行まではObject_No.2の共通の属性である、領域及び形状（外形、輪郭）に係る情報である。第21行から第23行はObject_No.2のシアン（C）の属性、色及び模様に係る情報である。第24行から第26行はObject_No.2のマゼンタ（M）の属性、色及び模様に係る情報である。Object_No.2にはイエロー（Y）とブラック（K）の情報は無い。

【0033】

（本実施形態の出力画像の描画例）

図10は、図9の前記display listの各パラメータと実際の描画Objectとの関係を示す図である。同図において文字 "A", "A", "A", "S", "U" の各1文字がひとつの描画objectの単位となっている。以下に、図9の値と図10の関係を簡単に説明する。図9はObject_No.1とObject_No.2しか記載されていないので、これらは図10の左の文字 "A", と次の文字 "A" に対応させて説明する。

【0034】

図9の第2行の描画高さ(Ha1)と幅(Wa1)とが、図10の最初の文字 "A" の高さ(Ha)と幅(Wa)とにそれぞれ対応する。ここでは、各文字の高さ及び幅は同じなので、Object_No.1の高さ(Ha1)及び幅(Wa1)を示す値とObject_No.2の高さ(Ha2)及び幅(Wa2)を示す値とは等しい。第3行の描画開始座標(Xa1,Ya1)は図10のXa,Yaである。更に、第4行のOffset座

10

20

30

40

50

標(Off_Xa1,Off_Ya1)は図10のOff_Xa1,Off_Ya1に相当する。すなわち、最小の文字 "A" の描画開始位置（領域の左上）は、図10の描画座標基準点からX方向（右方向）に(Xa + Off_Xa1)、Y方向（下方向）に(Ya + Off_Ya)である。ここで、Xa1=Xa2, Ya1=Ya2, Off_Ya1=Off_Ya2なので図10ではXa, Ya, Off_Yaと表示している。

【0035】

この描画開始位置(Xa + Off_Xa1, Ya + Off_Ya)から高さ（下方向）Ha, 幅（右方向）Waの矩形が属性の「領域」に相当する。

【0036】

第5行のobj._addr._pointer1の先のObject_HeaderからObj._Func.が文字（フォントの種類なども含む）を特定する情報である。幅(width)及び高さ(Height)は領域の幅及び高さと同じである。また、Obj._Body_addr._pointerが指し示すobjectの形状の圧縮dataとして文字"A"の圧縮データがObject_NO.1_codeに記憶されている。このObject_NO.1_codeが属性の「外形（輪郭）」に相当する。10

【0037】

そして、上述のColor_Gradiationが属性の「色」に、Fill_Patternが属性の「模様」に相当する。

【0038】

次のObject_No.2の文字"A"についても、上記の構成とほぼ同様である。但し、Object_No.1の文字"A"とは色が異なる。尚、図9には第18行のobj._addr._pointer2がObject_NO.2_codeを指し示すように図示されているが、図10のように同じ文字"A"の場合は、Object_NO.1_codeを指し示す。20

【0039】

（本実施形態における展開処理の分割例）

次に、本実施形態における展開処理をどのように分割して、複数のリコンフィギュラブルプロセッサで実行できるのかを説明する。

【0040】

本実施形態の展開処理は、基本的にライン単位である。すなわち、どの属性においても中間データはライン単位で一区切りとなるような構造になっている。従って、ページデータをライン単位で分割し、展開処理可能な構成である。その場合に、例えば「色(BG)」においては縦64pixel × 横64Pixelの矩形領域単位で分割が可能である。又、「輪郭(MSK)」の処理においては、line_skip機能が備わっている。このline_skip機能とは、例えば展開領域の上半分のラインと下半分のラインを分割しても、上半分のラインを読み飛ばして下半分のラインの先頭から展開を可能とする機能である。30

【0041】

以上説明したように、本実施形態ではライン単位の分割という規則による制限があるが、これは一例であり他の分割が可能な展開であればどのような処理でもよく、これらも本発明に含まれる。

【0042】

<本実施形態の描画処理回路の構成例>

図2は、図1におけるK, Y, M, C-plane用の描画処理回路であるRasterize_moduleの構成例を示したものである。前記各plane毎にこの図2で示す回路があり、計4個存在している。以下に各構成要素について説明を行う。40

【0043】

図2において、205に示されているRM1～RM4は、本実施形態で使用するリコンフィギュラブルプロセッサモジュール（再構成可能な処理回路モジュールの意）である。RMが行う処理については後述する。206は前記RM1～RM4からの出力データを格納するためのFIFOである。FIFO206は、RM1～RM4が描画領域の描画開始位置（通常、左上）からPixel単位（描画画素単位）に展開した出力データを順に記憶する。また、201は描画回路を再構成するためのデータを格納しているConfig._data_memoryである。202は前記データをRM1～RM4のどのリコンフィギュラブルプロセッサにloadするかを監視し、前記memoryからデータを読50

み出して各モジュールの回路再構成を制御している制御部(Config._control)である。尚、Config._data_memory201には、Config._control202がRM1～RM4をリコンフィギュアするスケジュールを管理するプログラムも格納されている。

【0044】

203はPDL(プリンタ記述言語)から変換された中間コード(display list; 図9参照)を解析して、描画Objectの属性毎にRM1～RM4で描画展開が可能なデータ(あるいは対応する中間コードそのままでもよい)を作成するDisplay_List解析部である。204は前記中間コードから各属性毎の描画展開に係る処理時間を予測し、後述の描画演算処理部207から進行状況のフィードバックをうけて、予測時間の補正を行う描画計測部である。描画演算処理部207は、前記RM1～RM4が展開した各属性毎の描画データをdisplay listに記載されている描画論理に従って、レイヤーのon/off、重ね合わせなどの演算をpixel単位で行う描画演算部である。208は前記の演算を行ったデータをプリンタエンジンに送出する前段のbufferであるBand_Mem.を示す。

【0045】

ここで、図1のBand_Mem.108は、少なくともBand_Mem.208が4plane分記憶できる容量が必要である。高速処理のためにバッファすることを考慮すると8plane分を用意するのが好適である。209は、図1のHard_Disk103などにオペレータによって予め格納してある図5の背景データ(例えば、フォームオーバーレイ、社標のロゴ、すかし文字、罫線など)を、展開処理されたホストからの描画データに重ね合わせるための加算器である。

【0046】

尚、FIFO206にRM1～RM4で展開処理したデータをどのように記憶し、描画演算処理部207でどのようにビット単位で合成するかについては、以下のような種々の方法があり、どの方法をとっても構わない。

- (1) RM1～RM4より出力するデータの先頭に属性とObject_Noを識別可能なタグを付けて識別する。
- (2) 図2のconfig._Control202からBPU207に対して描画Obj. No.と各RMで処理している属性の種類の対応を示す信号を出力する。
- (3) RMの後段にあるFIFO206を領域、輪郭、色、模様の各1つずつに割り当てておき、BPU207は例えばFIFO_1からFIFO_4までそれぞれから読み出したデータと前記属性と対応させる。

【0047】

又、図2では、Config._control202や、Display_List解析部203、描画計測部204、描画演算部207などを独立した構成要素として図示しているが、これらはCPUとmemoryとで構成された共通のコンピュータでソフト的に実現するものでよい。その場合に、Config._data_memory201や、CPUのプログラムがロードされるmemory、Display_Listをロードするmemoryなどは、1つのmemoryを共有してもよい。

【0048】

(リコンフィギュラブルプロセッサモジュール)

ここで、リコンフィギュラブルプロセッサモジュールについて簡単に説明を行う。

【0049】

リコンフィギュラブルプロセッサと称されるものは動的再構成(再構築)可能な論理装置である。これらは、機能分けされた専用回路ではなく、あらかじめ用意した複数の必要な回路の接続構成と動作設定をソフトウェアによって切り替えることで、専用回路と同等の機能を実現するものである。また、動作中にもそうした切り替えを可能とする特徴をも有するものがある。特に、動的に再構成が可能なりコンフィギュラブルプロセッサは、ダイナミックリコンフィギュラブルプロセッサと称されることが多い。

【0050】

一般的に、ダイナミックリコンフィギュラブルプロセッサは、チップ(論理装置)内に必要な回路を部分的にユニット化したハードウェア要素を予め複数準備する。そして、ユニットの選択や接続順序、動作設定といったパラメータやプログラムを事前にメモリに格

10

20

30

40

50

納しておき、その格納したプログラムやパラメータをソフトウェアによって動作中に瞬時に切り替えることにより、処理内容に応じて専用回路をもつチップのように振舞うものである。つまり、本実施形態のリコンフィギュラブルプロセッサ(RM)は、パラメータやプログラムを切り替えることで、たとえば、領域(AR)用描画回路として動作させた後に、輪郭(MSK)用描画回路として再構成させ、動作させることも可能となる。

【0051】

こうした思想に基づく技術は、FPGAなどに代表されるように古くから存在しているが、ダイナミックリコンフィギュラブルプロセッサは、集積度の向上や動的な切り替えが実現できる高速性において、FPGAよりも優れている。

【0052】

FPGAを構成するコンポーネントはゲートアレイであるため、実行できる処理も予め有する論理ゲート相当の機能となる。これに対してリコンフィギュラブルプロセッサを構成するユニットは小規模なプロセッサ相当か、プロセッサデータバス内の機能ブロック相当であるのが一般的となっている。そのため、当然ながらFPGAに比べてリコンフィギュラブルプロセッサの方が、高度な処理を実現可能である。

【0053】

なお、本発明を説明するためのいくつかの実施形態では、描画回路としてダイナミックリコンフィギュラブル技術を採用しているが、同様の処理が可能な構成であればこれに限定されない。

【0054】

<本実施形態の描画処理回路の動作例>

図3は、前記図1及び図2で示されるような画像出力制御装置において、各描画処理回路の再構成の手順例を示したフローチャートである。本実施形態では、この手順は各描画処理回路の独立したCPUで実行されるのが好ましいが、各構成要素がそれぞれのステップを実行する構成であっても、高速であって負担がなければ図1のCPU106が実行してもよい。

【0055】

まず、ステップS301において該当Band内の描画Objectの属性から図2の描画計測部204が1Bandの描画あたりに要する描画処理時間を見積もる。そして、ステップS302において各描画Object毎に描画回路での処理時間を予測する。この予測結果を基にして、ステップS303で描画回路をどのように再構成(リコンフィギュレーション)するかについての順番を、図2のconfig._Control部202でスケジューリングし、その情報(スケジューリングデータ)を図2のconfig._Data_memory201へ格納する。

【0056】

そして、ステップS304で、前記スケジューリングデータに従って1個目の描画Objectを描画するために必要な描画回路のコンフィギュレーションデータを図2のconfig._Data_memory201から読み出して、図2のRM1～RM4を構成する。そして、ステップS305で描画ハードウェアを起動する。ハードウェアが起動すると、コンフィギュレーションデータに基づいて構成されたRM1～RM4が各描画Object毎の属性に応じた描画データの展開を開始し、展開が終了したデータを図2のFIFO206へ順次格納していく。

【0057】

また、ステップS306では、RM1～RM4の各moduleからのステータス信号を受けて、描画処理が早めに終了しているものがないかを判定する。なお、このステップは図2の描画計測部204が実行する。その後、ステップS307では、前記描画計測部204からの予測情報と現状の処理ステータス(描画実行中 or 描画終了)を受けて、未だ描画処理に時間がかかるようなobjectの属性のものがあるか否かを判断する。そして、特定の属性のものが他の属性に比べて処理時間がかかると予測される場合には、ステップS308で負荷を分散するように図2のconfig._Data_memory201に格納されているスケジューリングデータを修正し、これに基づいてステップS304からの処理を再度行う。

【0058】

10

20

30

40

50

一方、ステップS306で描画処理が早めに終了しているものが無い、またはステップS307でボトルネックとなるような属性を持った描画Objectが無いと判断された場合には、ステップS309で描画論理演算(BPU)を実行し、当該描画objectの処理を終了する。その後、ステップS311で現在描画処理をしているBandにおいて全ての描画Objectの展開が終了するまでチェックをし、繰り返し前記の処理を行うものである。

【0059】

(本実施形態の描画処理のタイミング例)

図4の下段は、図3のフローチャートに従ってスケジューリングを修正して描画回路(RM)の再構成を行う場合について、従来の固定した回路構成での処理時間(図4の上段)と比較して示したものである。RM1～RM4は、描画Objectの属性である領域データ(AR)、輪郭データ(MSK)、色データ(BG)、模様データ(GD)を並列で処理するためのリコンフィギュラブルな回路モジュールである。タイミングチャートの横軸が、各RM1～RM4が各自の処理に要する時間を示している。10

【0060】

図4の上段に記載している従来例においては、描画Object1を描画する際ににおいて、例えばRM1やRM2の「領域(AR)」と「輪郭(MSK)」の処理が早く終了しても、RM4の「模様(GD)」の処理が終わるまでは図2の描画演算モジュール: Bit Processing Unit207で次のObject2の描画処理を開始できないため、待たされる状態(idle状態)になっていた。

【0061】

一方、図4の下段に記載している本実施形態の例では、従来例で描画処理が早く終了したRM1, RM2などのモジュールについては、回路の再構成が行われたことによって、描画Object1の展開処理が従来例よりも早く終了し、最終的に描画Object2についての描画演算モジュール(BPU)の処理を早く開始することが可能となる。以下、描画Object 2,3,4...についても同様である。20

【0062】

このように、本発明を適用することにより、描画Object1～4までに要する時間が従来の固定した処理回路の場合と比較して処理時間を大幅に短縮することが可能となる。

【0063】

具体的には、PDLを解析して中間コード(display list)に変換した後の描画処理をハードウェアの固定した処理によって行うのではなく、実際の回路の処理時間と処理に必要なデータサイズを鑑みて動的に回路構成を変更することにより、1 Bandあたりの描画ObjectをBit Map Dataへ展開するために要する処理時間を短縮させるという効果がある。30

【0064】

尚、図4の下段の再構成はかなりの時間幅で表示しているが、実際の再構成の処理はシステムクロックの1 cycleで実現でき、全体の処理にはほとんど影響しない。RM1～RM4の再構成は、ソフトプログラムとパラメータを入れ替えるものであっても、一部ハードウェアの選択や繋ぎ替えなどを含んでもよい。

【0065】

<本実施形態の描画処理回路の他の動作例>

上記実施形態においては、描画回路を再構成する際の判断基準として各属性の描画が終了したという実時間の処理状態を元にしてスケジューリング、あるいは見積もり時間の補正をしていた。本実施形態の動作例では予め各属性に係る描画処理時間を見積もった後で、負荷(時間)のかかる属性の回路を優先的に再構成をするというものである。40

【0066】

図6は、本実施形態の処理手順を示すフローチャートであり、以下順を追って処理を説明する。

【0067】

まず、ステップS601で描画Objectの属性から描画処理時間を見積もり、ステップS602で各Objectの属性毎の処理時間をtotalデータとして格納しておく。

【0068】

50

図7は、図6のステップS602で格納しているtotalデータの例である。図7において、1 Band目はN個の描画Objectが存在し、2 Band目にはM個の描画Objectが存在するものとする。各Object毎の描画処理時間の予測値を格納した後で、各BandにおいてA.領域、B.輪郭、C.色、D.模様の各属性毎の描画処理時間の合計を示すtotal_time（実際はsystem clockのサイクル数で表す）についても予測を行う。次に、前記のtotalデータを基にして、ステップS603で描画回路を再構成する順番をスケジューリングし、ステップS604で前記totalデータに格納されているtotal_timeの大きい属性のものから優先的に回路の再構成を行う順番を決定する。

【0069】

そして、この結果は、図2のConfig._data_memory201へ格納された後に、ステップS605で描画ハードウェアが起動される。ハードウェアが起動されると、RM1～RM4が各描画Object毎の属性に応じた描画データの展開処理を開始し、展開が終了したデータを図2のFIFO206へ順次格納していく。また、図2の描画計測部204はRM1～RM4までの各moduleからのステータス信号を受けて、ステップS606で描画処理が早めに終了しているものがないかをセンスする。その後、前記描画計測部204からの予測情報と現状の処理ステータス（描画実行中 or 描画終了）を受けて、ステップS607で現在のBandにおいて描画処理のtotal_time = 0 (system cycle)のものが存在するか否かを判断する。ここで、total_time = 0となる属性のものが存在する場合には、ステップS608で負荷を分散するように図2のconfig. Data memory201に格納されているスケジューリングデータを修正し、これに基づいてステップS604からの処理を再度行う。

10

【0070】

一方、ステップS606で描画処理が早めに終了しているものが無い、またはステップS607でボトルネックとなるような属性を持った描画Objectが無いと判断された場合には、ステップS609でBPUの描画論理演算を実行し、当該描画objectの処理を終了する。その後、ステップS611で現在描画処理をしているBandにおいて全ての描画Objectの展開が終了するまでチェックをし、繰り返し前記の処理を行うものである。

20

【0071】

（本実施形態の描画処理のタイミング例）

図8は、図6のフローチャートにともづいて、図2のRP1～RP4までの回路を再構成したときの例を示したものである。

30

【0072】

先の実施形態で示した図4との違いは、obj.1の処理において最も処理時間を要する属性である「模様(GD)」「色(BG)」から再構成を行い、それが終了した後で属性「領域(AR)」「輪郭(MSK)」の処理回路を再構成している点である。

【0073】

このように再構成のスケジューリングを実行することで、さらに展開処理を高速に実行することができる。

【図面の簡単な説明】

【0074】

【図1】本実施形態に係る画像出力制御装置の構成を示す図である。

40

【図2】本実施形態に係る画像出力制御装置におけるRasterize_moduleの構成を示す図である。

【図3】本実施形態に係る画像出力制御装置の処理の第1例の処理手順を説明するフローチャートである。

【図4】本実施形態に係わる描画処理回路の再構成タイミングの第1例を示す図である。

【図5】本実施形態に係わる描画処理回路で展開する描画Objectの属性を示す図である。

【図6】本実施形態に係る画像出力制御装置の第2例の処理手順を説明するフローチャートである。

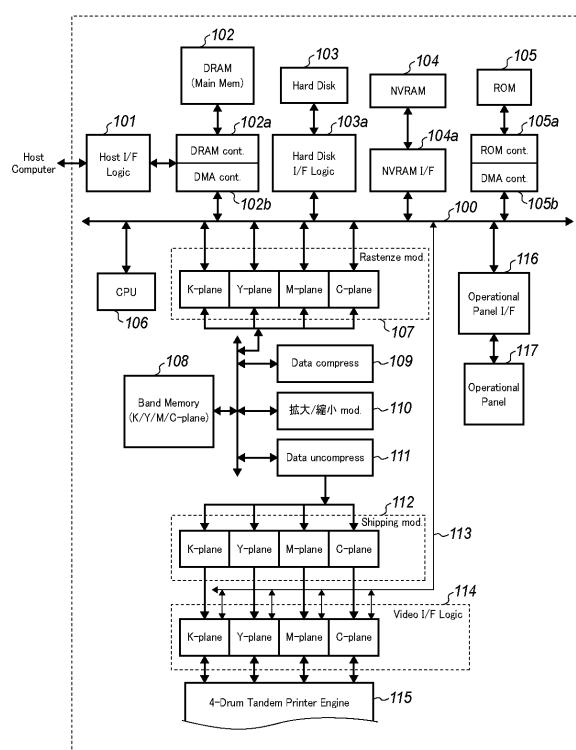
【図7】本実施形態に係る描画処理回路の描画処理時間を格納するtableデータの構成を示す図である。

50

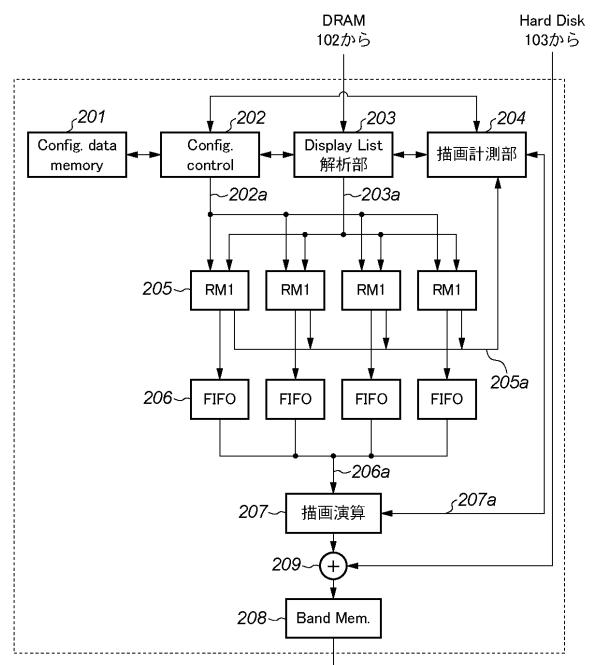
【図8】本実施形態に係わる描画処理回路の第2例の再構成タイミングを示す図である。
 【図9】本実施形態に係わる描画処理回路がdecodeする中間コード(display list)の例を示す図である。

【図10】本実施形態に係わる描画objectと描画座標の関係を示す図である。

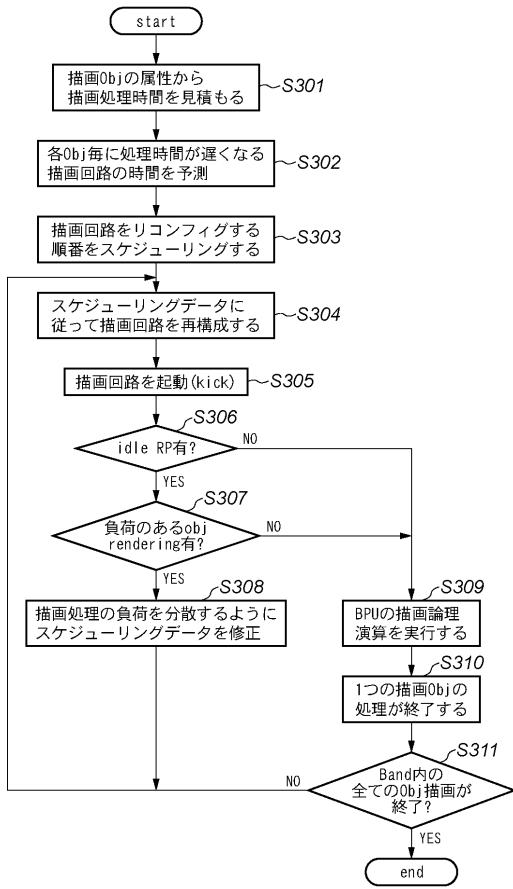
【図1】



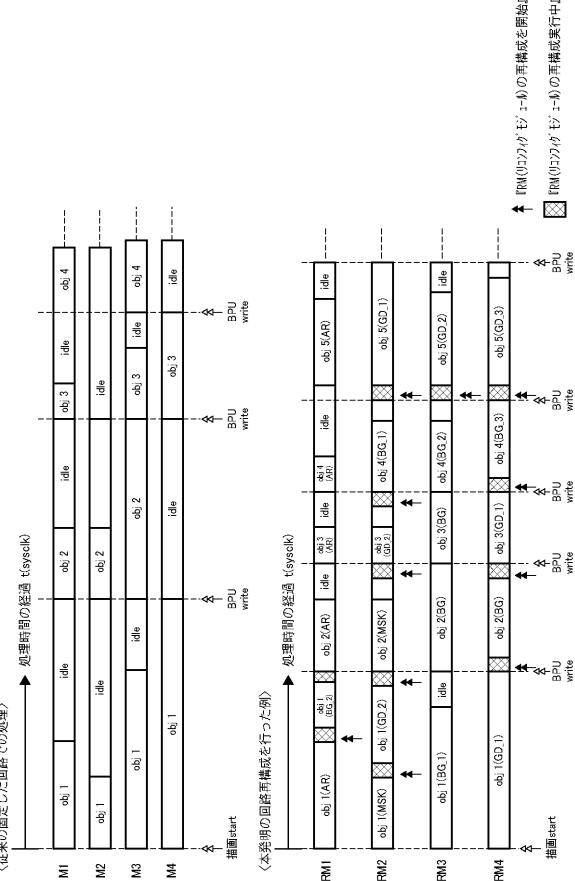
【図2】



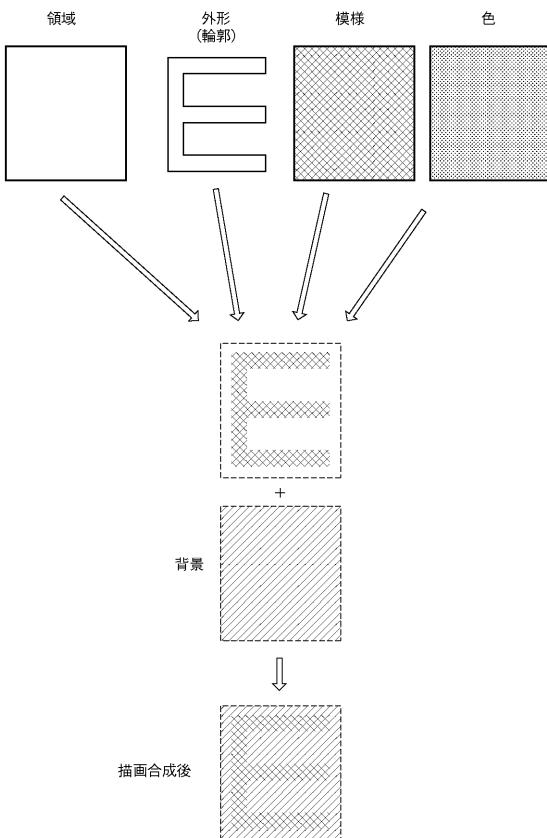
【図3】



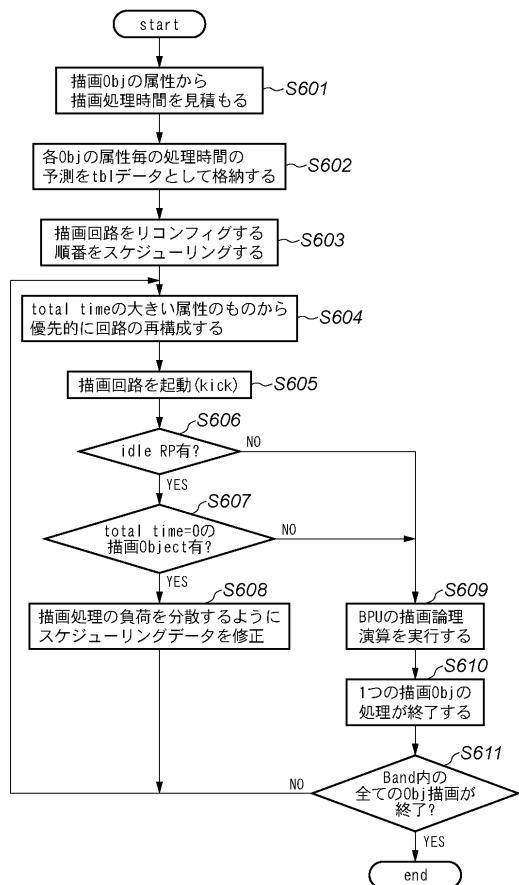
【図4】



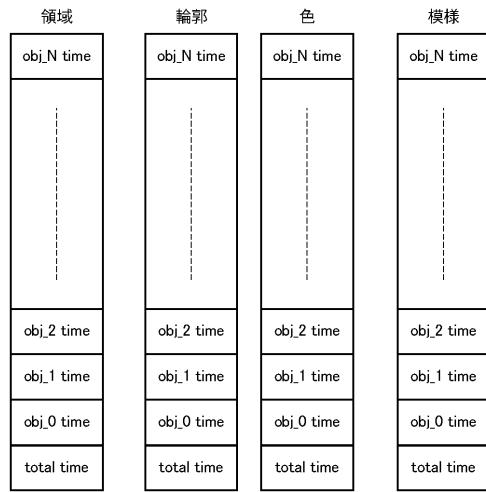
【図5】



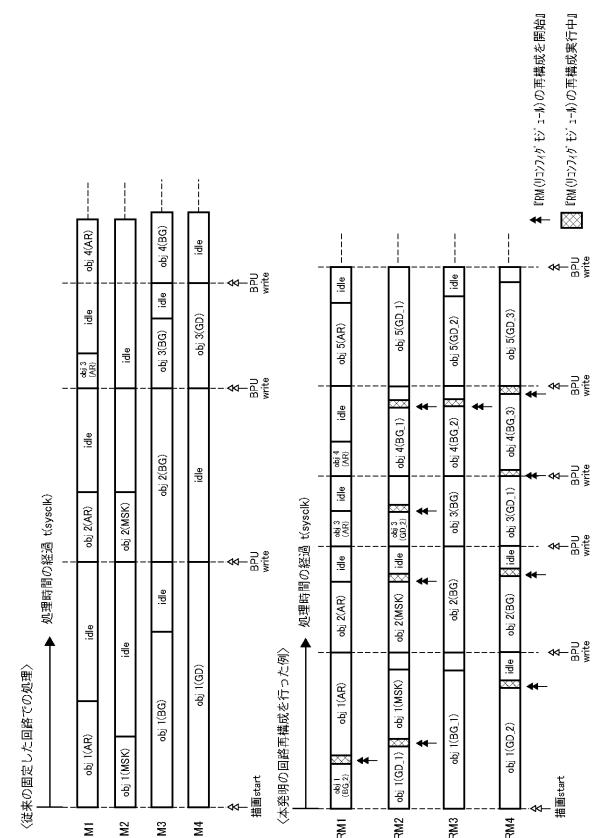
【図6】



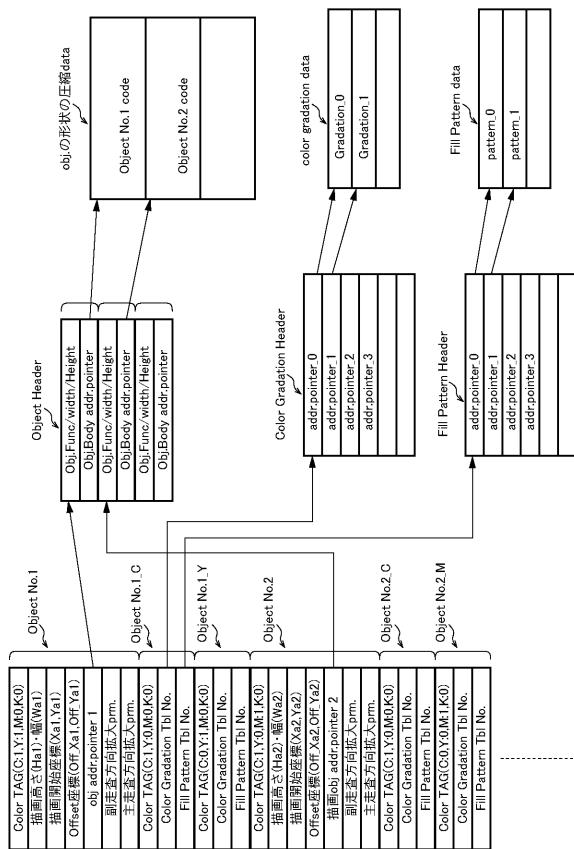
【図7】



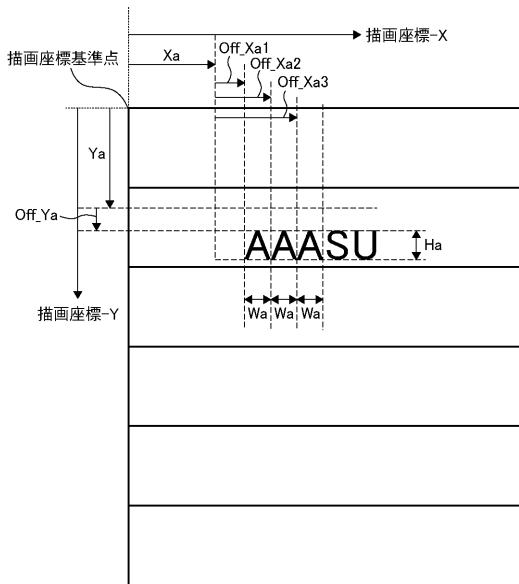
【図8】



【図9】



【図10】



フロントページの続き

(72)発明者 立川 知弘

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

審査官 松川 直樹

(56)参考文献 特開2001-010123(JP,A)

特開2000-351240(JP,A)

特開平11-144062(JP,A)

(58)調査した分野(Int.Cl., DB名)

B 41 J 5 / 30

G 06 F 3 / 12